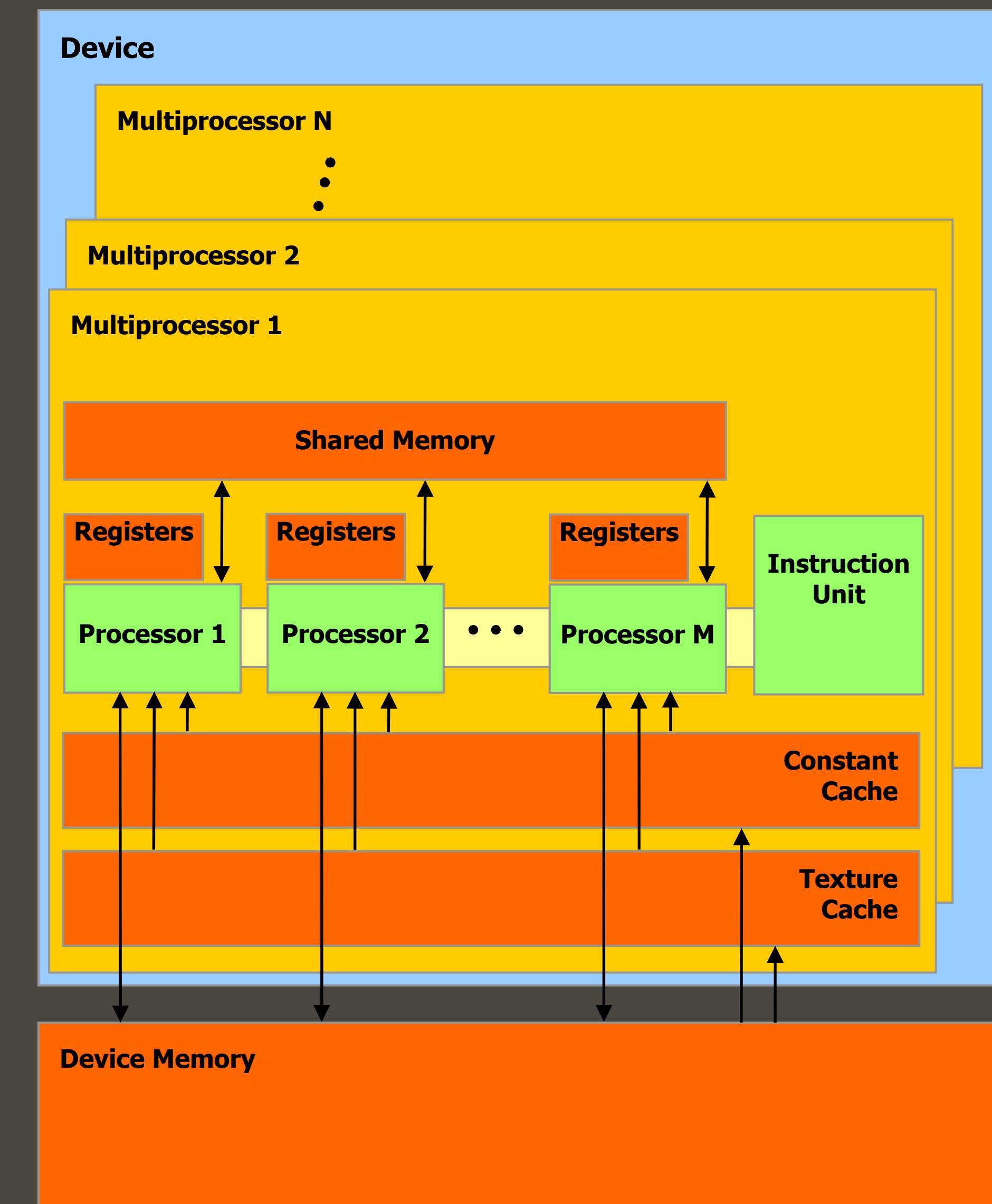
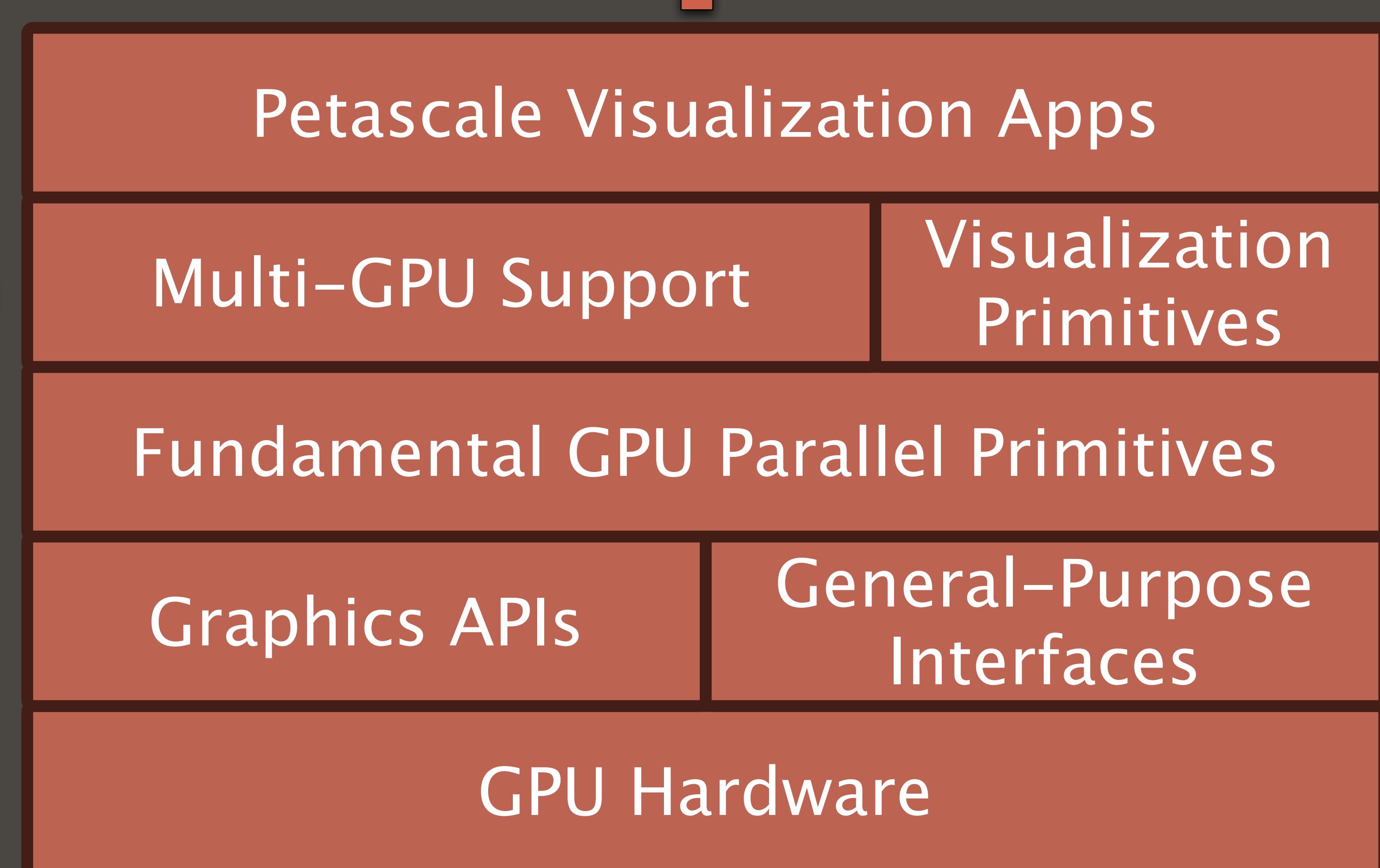
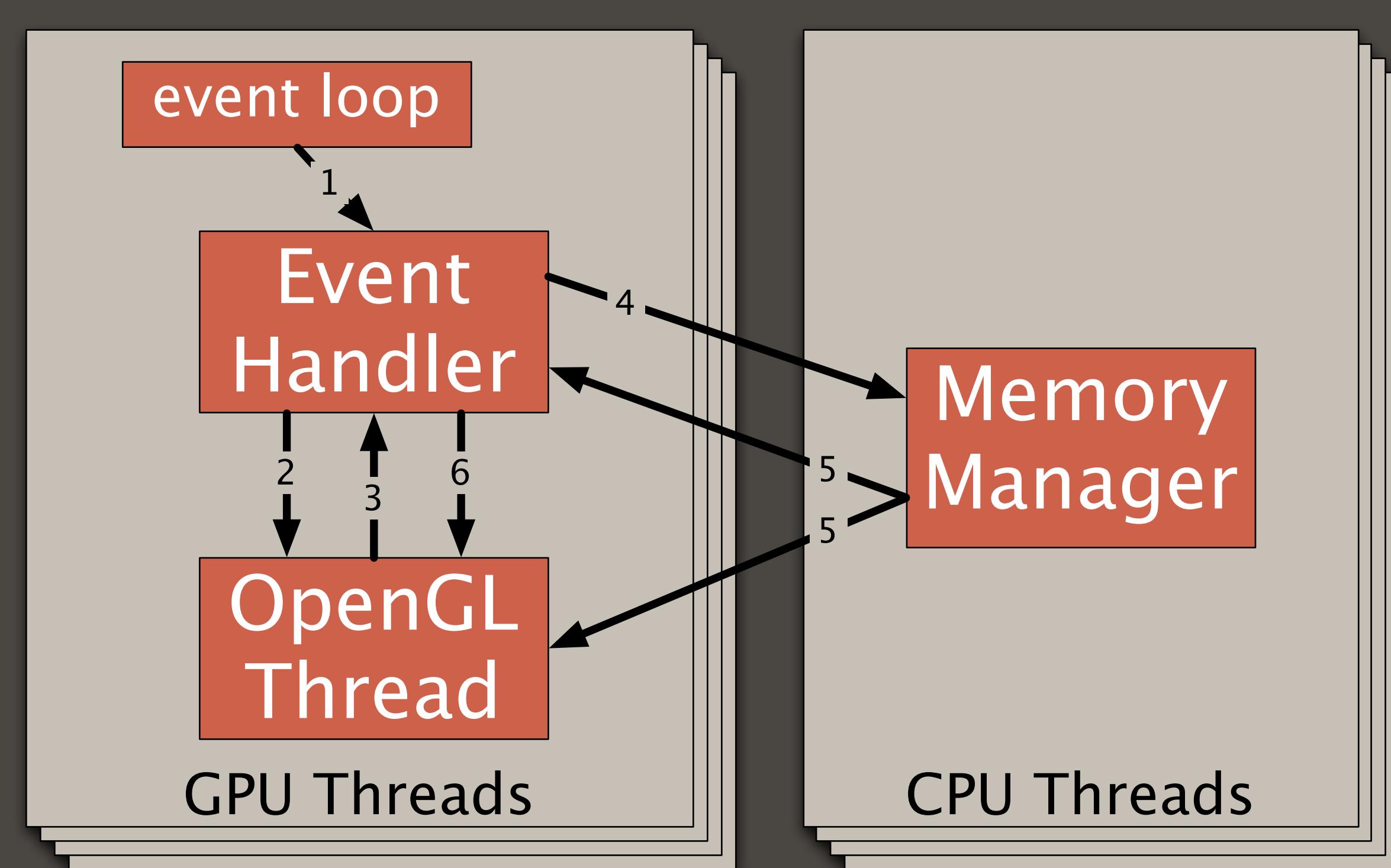
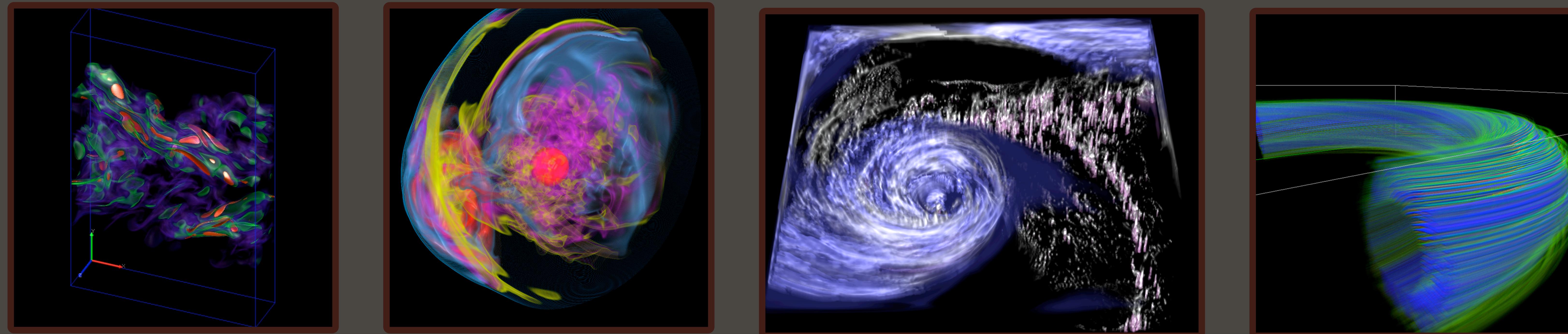


# Towards Multi-GPU Support for Visualization

John Owens  
University of California at Davis

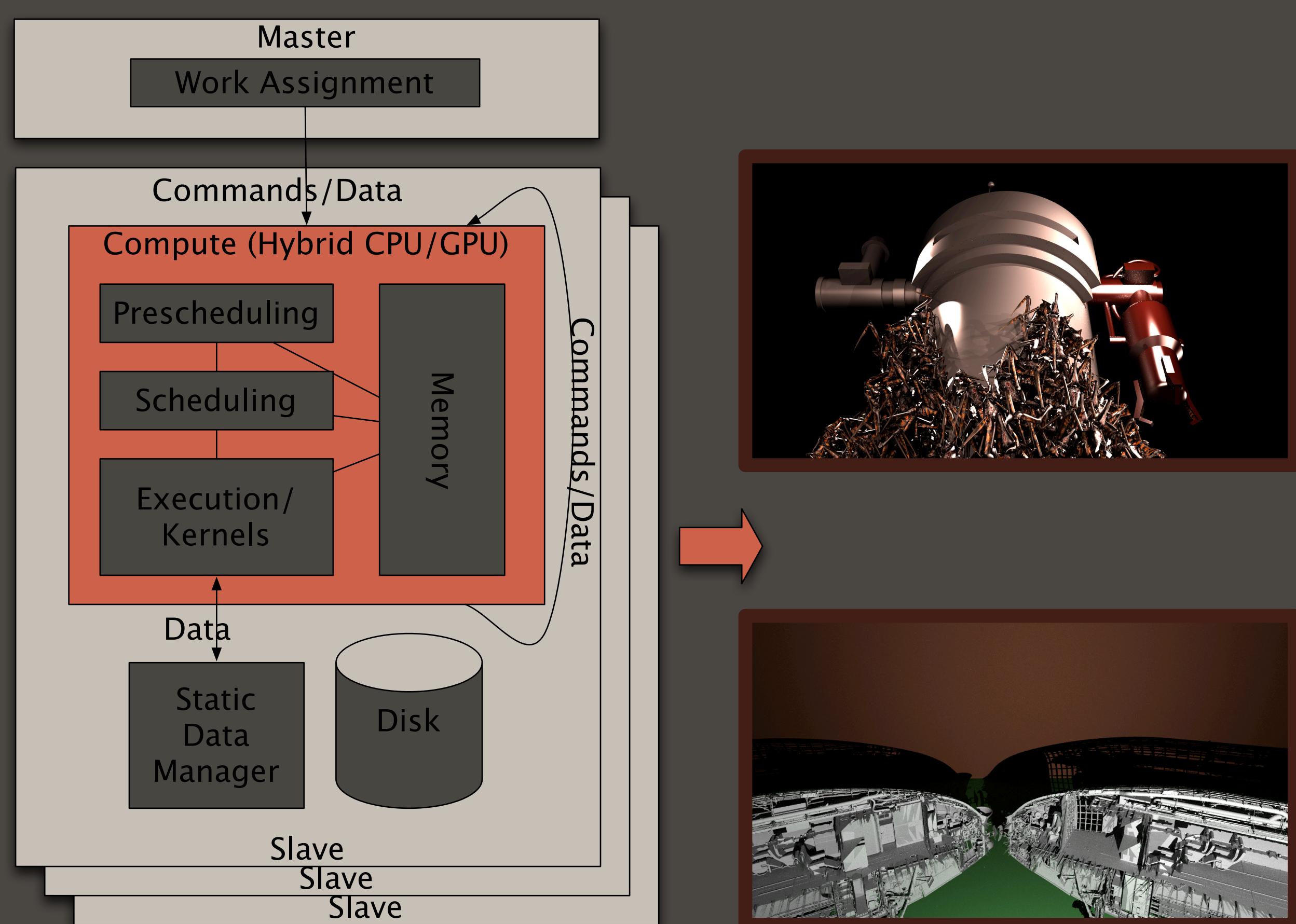


**General-purpose Interfaces**

- New GPU programming interfaces like NVIDIA's CUDA (above) and AMD's CTM allow direct programmer access to the high-performance parallel processors in the GPU.

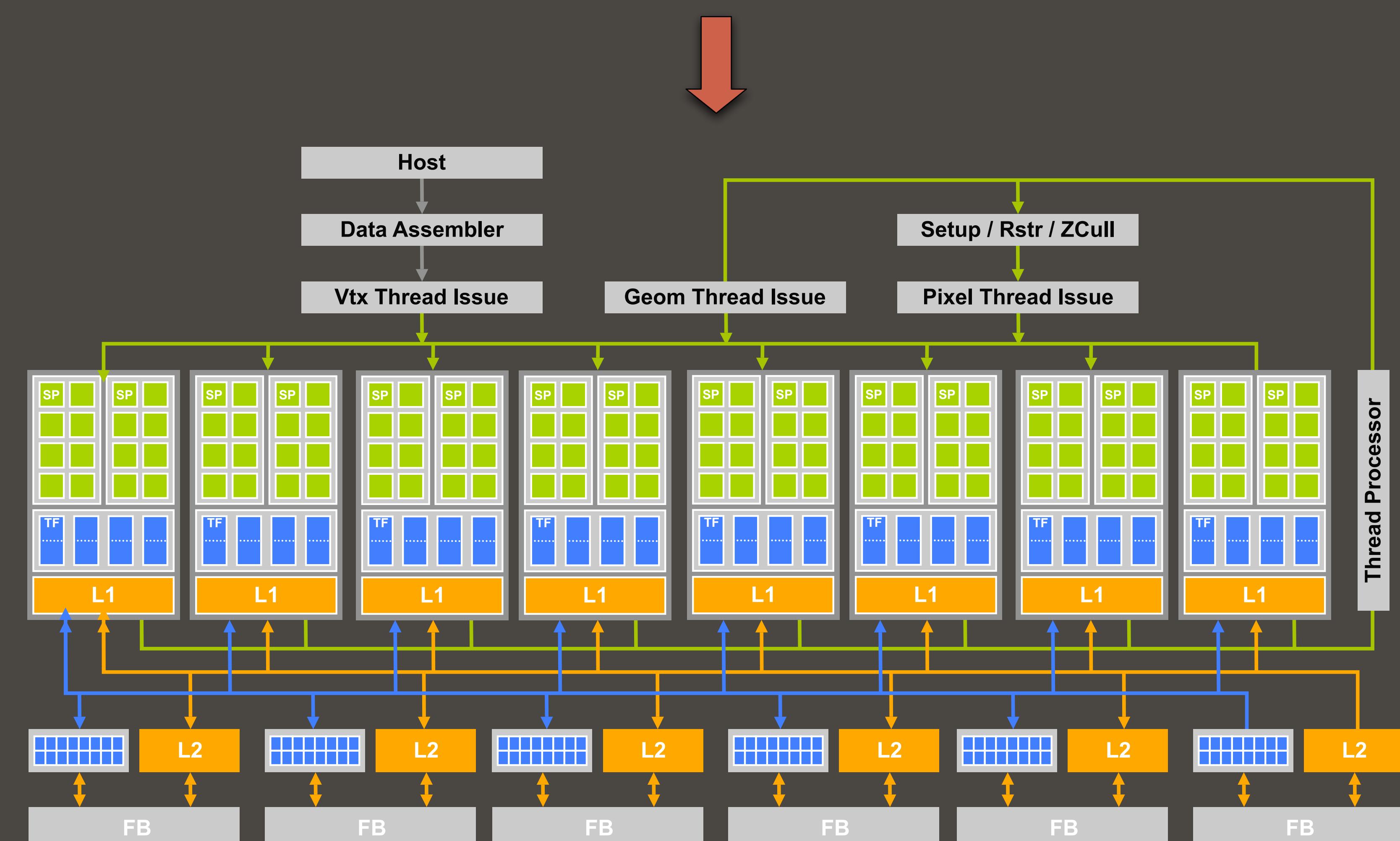
**Multi-GPU Distributed Memory**

- A distributed-shared-memory abstraction across GPUs divides global memory across the GPUs; with our DSM implementation, any GPU can read or write data to or from any memory location.



**Hybrid CPU-GPU Computation**

- A complex out-of-core application like visualization or (above) high-quality rendering can effectively divide its work between CPUs and GPUs, using the GPU for compute-heavy, parallel-friendly tasks.



**Today's GPU**

- Modern GPUs are organized around large parallel arrays of thread processors (the NVIDIA GeForce 8800 GTX above has 128 processors) and feature superior performance and price-performance (right) compared to CPUs.

